Claims

1. A multiple modulus conversion (MMC) method for obtaining a plurality of index values associated with a plurality of moduli, for use in a communication system having means configured to map frames of information bits onto predetermined communication signal parameters, said method comprising:

obtaining an input;

representing the input as a plurality of sub-quotients;

performing a multiplication operation to multiply at least one of the sub-quotients by a multiplicand; and

determining an index value associated with the multiplicand, the index value being responsive to the inverse modulus multiplication operation.

2. A method according to Claim 1, further comprising obtaining a multiplicand that relates to the estimated inverse of a whole number, wherein performing a multiplication operation includes multiplying at least one of the sub-quotients by the multiplicand.

3. A method according to Claim 2, wherein at least one pseudo remainder is produced, the method further comprising checking at least one of the pseudo remainders to determine whether the pseudo remainder falls within a predetermined range and, if it is not within the predetermined range, adjusting the pseudo remainder such that the pseudo remainder falls within the predetermined range.

4. A method according to Claim 2, wherein at least one pseudo remainder is produced, the method further comprising checking at least one of the pseudo remainders to determine whether the pseudo remainder is larger than or equal to a predetermined number and, if it is, subtracting the said number from the pseudo remainder.

5. A method according to Claim 4, wherein the predetermined number is subtracted from the pseudo remainder until the pseudo remainder is smaller than the predetermined number.

6. A method according to Claim 2, wherein at least one of the multiplicands is pre-calculated, and stored in a table, the method further comprising obtaining at least one multiplicand from the table.

7. A method according to Claim 2, wherein the maximum number of digits N_3 in the multiplicands, the maximum number of digits N_2 in the whole numbers, and the maximum number of digits n(j) in all but the most significant of the sub-quotients are related by the inequality $N_3 > N_2 + n(j)$.

8. A method according to Claim 2, wherein the multiplicand is calculated according to the following formula: $C_i = \text{floor}(B^{N3}/Y_i)$, where B is the base of the numbering system, N_3 is the maximum number of digits in the multiplicands, and Y_i is the whole number.

9. A method according to Claim 2, wherein the input is represented as sub-quotients in the form of $Q_{i-1} = Q_{i-1,0} + Q_{i-1,1}*B^{n(0)} + \ldots + Q_{i-1,k}*B^{n(0)+n(1)+..n(k-1)}$.

10. A method according to Claim 8, wherein a sub-quotient of the output pseudo-quotient is obtained by multiplying a portion of the input by the multiplicand C_i according to the following formula: $Q_{i,j} = ((Q_{i-1,j}+R_{i,j+1}*B^{n(j)})*C_i)>>N_3$; and wherein the output pseudo-remainder of the sub-quotient calculation is obtained according to the following formula: $R_{i,j}=(Q_{i-1,j}+R_{i,j+1}*B^{n(j)})-(Q_{i,j}*Y_i)$, where B is the base number system.

11. A method according to Claim 8, further comprising producing an output remainder R_i and new quotient $Q_i = Q_{i,0} + Q_{i,1}*B^{n(0)} + \ldots + Q_{i,k}*B^{n(0)+n(1)+..n(k-1)}$.

12. A multiple modulus conversion (MMC) encoder comprising:

an input configured to receive input data represented by a number of digits;

a multiplication means for multiplying two numbers; and

a processor configured to represent the input as a plurality of sub-quotients, to multiply at least one sub-quotient by a multiplicand, and to produce an index value associated with the multiplicand, the index value being responsive to the multiplication operation.

13. A modulus encoder according to Claim 12, wherein the multiplicand is related to the estimated inverse of a whole number.

14. A modulus encoder according to Claim 13, wherein the processor is further configured to generate a pseudo remainder, and further configured to check whether the pseudo remainder falls within a predetermined range.

15. A modulus encoder according to Claim 13, wherein the processor is further configured to generate a pseudo remainder, and further configured to check whether the pseudo remainder is larger than or equal to a predetermined number, and further configured to subtract the predetermined number from the pseudo remainder if the pseudo remainder is larger than the predetermined number.

16. A modulus encoder according to Claim 15, wherein the processor is configured to keep subtracting the predetermined number from the pseudo remainder until the pseudo remainder is smaller than the predetermined number.

17. A modulus encoder according to Claim 13, further including storage means to store at least one multiplicand.

18. A modulus encoder according to Claim 13, wherein the processor is configured to handle multiplicands with a maximum number of digits N_3, and is further configured to handle moduli with a maximum number of digits N_2, and is further configured to handle inputs represented as sub-quotients with a maximum number of

digits n(j) in all but the most significant of the sub-quotients, wherein these quantities are related by the inequality $N_3 > N_2 + n(j)$.

19. A modulus encoder according to Claim 13, further including a inverse estimate calculation means, wherein the inverse estimate is calculated according to the following formula: $C_i = \mathrm{floor}(B^{N3}/Y_i)$, where B is the base of the number system, N_3 is the maximum number of digits in the multiplicands, and Y_i is a whole number.

20. A modulus encoder according to Claim 13, wherein the input is configured to obtain an input value in the form $Q_{i-1} = Q_{i-1,0} + Q_{i-1,1}*B^{n(0)} + \ldots + Q_{i-1,k}*B^{n(0)+n(1)+..n(k-1)}$.

21. A modulus encoder according to Claim 18, wherein the processor is configured to obtain a sub-quotient of an output pseudo-quotient by multiplying a portion of the input by an estimated inverse modulus according to the following formula: $Q_{i,j} = ((Q_{i-1,j}+R_{i,j+1}*B^{n(j)})*C_i)>>N_3$; and wherein the processor is further configured to obtain a pseudo remainder according to the following formula: $R_{i,j}=(Q_{i-1,j}+R_{i,j+1}*B^{n(j)})-(Q_{i,j}*Y_i)$, where B is the base of the numbering system.

22. A modulus encoder according to Claim 18, wherein the processor is further configured to generate an output remainder R_i and new quotient $Q_i = Q_{i,0} + Q_{i,1}*B^{n(0)} + \ldots + Q_{i,k}*B^{n(0)+n(1)+..n(k-1)}$.